

Claims

1. A scalable agent service system that supports an arbitrary number of computer software agents for providing services or information to an arbitrary number of client computation devices, comprising:

one or more adaptive engines that receive and apply inferencing to external information received from one or more data feeds to generate computer-implemented tasks relating to the external information; and

one or more service fulfillment engines that operate asynchronously with the one or more adaptive engines to perform operations in accordance with the computer-implemented tasks.

2. The system of claim 1 in which the one or more adaptive engines each include plural computer-implemented event agents, one or more of which operate as rules engines that apply the inferencing to the external information.

3. The system of claim 1 in which the one or more adaptive engines each include a metadata repository for temporarily storing the external information as metadata.

4. The system of claim 3 in which the one or more adaptive engines include plural computer-implemented event agents, one or more of which operate as rules engines that retrieve the external information from the metadata repository and apply the inferencing to the external information.

5. The system of claim 4 in which the one or more adaptive engines each include one or more data managers that place the external information in the metadata repository asynchronously relative to operation of the event agents.

6. The system of claim 3 in which the one or more adaptive engines each include an isochronal agent that periodically analyses the metadata in the metadata repository at predefined time intervals to generate corresponding computer-implemented tasks.

7. The system of claim 2 in which the one or more adaptive engines include relational databases for storing one or more rules that are retrieved and

used by the event agents that operate as rules engines to apply the inferencing to the external information.

8. The system of claim 7 in which the relational database further stores events that correspond to tasks to be performed at predefined times, the one or more adaptive engines each further including a date/time daemon that scans the relational database to identify events that are scheduled during each time period and generating corresponding tasks to be performed.

9. The system of claim 1 in which the computer-implemented tasks are ordered in one or more task queues and the one or more adaptive engines include:

dispatchers that retrieve the computer-implemented tasks from the one or more task queues; and

service agents that receive the computer-implemented tasks from the dispatchers to process and forward the computer-implemented tasks to the one or more service fulfillment engines.

10. The system of claim 9 in which the one or more adaptive engines include pools of plural instances of service agents, each computer-implemented task being executed on a separate processing thread with a corresponding instance of a service agent.

11. The system of claim 1 in which the operations performed in accordance with the computer-implemented tasks by the one or more service fulfillment engines include providing information to the client computation devices in any of plural communication formats.

12. The system of claim 11 in which the one or more service fulfillment engines include separate service task queues for each of the plural communication formats.

13. The system of claim 12 in which the one or more service fulfillment engines include separate dispatchers that manage the separate service task queues for each of the plural communication formats.

14. In a computer readable medium, scalable agent service software that supports an arbitrary number of computer software agents for providing services or information to an arbitrary number of client computation devices, comprising:

adaptive engine software that receives and applies inferencing to external information received from one or more data feeds to generate computer-implemented tasks relating to the external information; and

service fulfillment engine software that operates asynchronously with the adaptive engine software to perform operations in accordance with the computer-implemented tasks.

15. The medium of claim 14 in which the adaptive engine software includes plural computer-implemented event agents, one or more of which operate as rules engines that apply the inferencing to the external information.

16. The medium of claim 14 in which the adaptive engine software includes metadata repository software for temporarily storing the external information as metadata.

17. The medium of claim 16 in which the adaptive engine software includes plural computer-implemented event agents, one or more of which operate as rules engines that retrieve the external information from the metadata repository software and apply the inferencing to the external information.

18. The medium of claim 17 in which the adaptive engine software includes data manager software that provides the external information in the metadata repository software asynchronously relative to operation of the event agents.

19. The medium of claim 16 in which the adaptive engine software includes an isochronal agent that periodically analyses the metadata of the metadata repository software at predefined time intervals to generate corresponding computer-implemented tasks.

20. The medium of claim 15 in which the adaptive engine software includes a relational database for storing plural rules that are retrieved and used by the event agents that operate as rules engines to apply the inferencing to the external information.

21. The medium of claim 20 in which the relational database further stores events that correspond to tasks to be performed at predefined times, the adaptive engine software further including a date/time daemon that scans the relational database to identify events that are scheduled during each time period and generating corresponding tasks to be performed.

22. The medium of claim 20 in which the adaptive engine software further includes:

software for splitting the relational database into key-range partitions; and
software for activating plural instances of service agents that service corresponding key-range partitions of the relational database.

23. The medium of claim 22 in which the plural instances of service agents are activated concurrently to service the corresponding key-range partitions of the relational database.

24. A scalable agent service method that supports an arbitrary number of computer software agents for providing services or information to an arbitrary number of client computation devices, comprising:

receiving and applying inferencing with plural computer-implemented event agents to external information received from one or more data feeds to generate computer-implemented tasks relating to the external information, one or more of the event agents operating as rules engines based upon rules stored in a relational database that is split into plural key-range partitions;

activating plural instances of service agents that service corresponding key-range partitions of the relational database to obtain service operations; and
performing the service operations asynchronously with the receiving and applying of inferencing to the external information.

25. The method of claim 24 in which the plural instances of service agents are activated concurrently to service the corresponding key-range partitions of the relational database.

26. The method of claim 25 in which the relational database is split into N-number of key-range partitions and N-number of instances of the service agents are activated concurrently to service the corresponding key-range partitions.

27. The method of claim 24 in which subscribers utilize the method and are assigned keys corresponding to the key-range partitions, the method further comprising assigning the keys to the subscribers to distribute them generally uniformly across the key-range partitions.

28. A scalable agent service system adaptive engine that supports an arbitrary number of computer software agents for providing services or information to an arbitrary number of client computation devices, comprising:

one or more input interfaces for receiving information from one or more external data feeds;

plural computer-implemented event agents, one or more of which operate as rules engines that apply inferencing to the external information;

a relational database for storing one or more rules that are retrieved and used by the event agents that operate as rules engines to apply the inferencing to the external information; and

event scheduler means for scheduling at least first and second different types of asynchronous events for execution by the event agents as computer-implemented tasks.

29. The system of claim 28 in which the first and second different types of asynchronous events include any two of periodic events, non-periodic scheduled events, and spontaneous events.

30. The system of claim 28 in which one of the first and second different types of asynchronous events are non-periodic scheduled events, the system further comprising a date/time daemon that scans the relational database to identify events that are scheduled during each time period and generating corresponding tasks for executing the events.

31. The system of claim 28 in which the computer-implemented tasks are ordered in one or more task queues, the system further comprising:

dispatchers that retrieve the computer-implemented tasks from the one or more task queues; and

service agents that receive the computer-implemented tasks from the dispatchers to process the computer-implemented tasks.

32. The system of claim 28 in which one of the first and second different types of asynchronous events are spontaneous events and in which a computer-implemented event agent designates the spontaneous events as receiving immediate or delayed processing according to a predefined rule.

33. The system of claim 32 in which spontaneous events designated as receiving delayed processing are processed together in batches of the same.

34. The system of claim 32 in which the other of the first and second different types of asynchronous events are periodic events and in which the spontaneous events designated as receiving delayed processing are processed concurrently with the periodic events.

35. A scalable agent service system adaptive engine method that supports an arbitrary number of computer software agents for providing services or information to an arbitrary number of client computation devices, the method comprising:

receiving information at one or more input interfaces from one or more external data feeds;

activating plural computer-implemented event agents, one or more of which operate as rules engines that apply inferencing to the external information;

storing in a relational database one or more rules that are retrieved and used by the event agents that operate as rules engines to apply the inferencing to the external information; and

scheduling at least first and second different types of asynchronous events for execution by the event agents as computer-implemented tasks.

36. The method of claim 35 in which the first and second different types of asynchronous events include any two of periodic events, non-periodic scheduled events, and spontaneous events.

37. The method of claim 35 in which one of the first and second different types of asynchronous events are non-periodic scheduled events, the method further comprising a scanning the relational database to identify events that are scheduled during each time period and generating corresponding tasks for executing the events.

38. The method of claim 35 further comprising:
ordering the computer-implemented tasks in one or more task queues;
retrieving the computer-implemented tasks from the one or more task queues; and

providing the computer-implemented tasks to service agents to process the computer-implemented tasks.

39. The method of claim 35 in which one of the first and second different types of asynchronous events are spontaneous events and in which the method includes designating the spontaneous events as receiving immediate or delayed processing according to a predefined rule.

40. The method of claim 39 further comprising processing the spontaneous events designated as receiving delayed processing together in batches of the same.

41. The method of claim 39 in which the other of the first and second different types of asynchronous events are periodic events, the method further including processing the spontaneous events designated as receiving delayed processing concurrently with the periodic events.